

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A method of arranging objects comprising:
setting a class hierarchy, wherein
the class hierarchy comprises an upper level class and a lower level class,
and
the objects are members of at least one of the upper level class and the
lower level class;
assigning an attribute to the **[[top]]** upper level class, wherein the attribute
describes the objects; and
inheriting of the attribute by the lower level class, wherein
the attribute is within a first domain with regard to the upper level class,
the attribute is within a second domain with regard to the lower level class,
a second domain value set of the second domain is smaller than a first
domain value set of the first domain, and
the attribute is restricted to the second domain value set upon the
inheriting.
2. (Currently Amended) The method of arranging objects of claim 1, further
comprising:
superceding said attribute of said upper level class by assigning an attribute to the
lower level class, the attribute describing an object that is a member of the
lower level class.
3. (Original) The method of arranging objects of claim 1, wherein the
attribute comprises a distinctive domain value set.
4. (Original) The method of arranging objects of claim 1, wherein the class
hierarchy further comprises a class below the lower level class in the class hierarchy, and
further comprising:

inheriting of the attribute by the class.

5. (Original) The method of arranging objects of claim 1, further comprising:

expanding the class hierarchy horizontally by adding a class to the lower level class; and
inheriting of the attribute by the class.

6. (Currently Amended) A hierarchical class architecture of objects stored in a memory comprising:

an upper level class;
a lower level class, wherein the upper and lower level classes are stored in the memory; **[[and]]**
a first domain value set of a first domain of the upper level class;
a second domain value set of a second domain; and
an attribute, wherein
the attribute is assigned to the upper level class,
the attribute is within **[[a]]** the first domain ~~with regard to the upper level~~
class,
the attribute is within the second domain,
the objects are members of at least one of the upper level class and the lower level class,
the attribute describes the objects, **[[and]]**
the lower level class is configured to inherit the attribute,
the first domain value set is smaller than the second domain value set, and
the attribute is restricted to the second domain value set upon the attribute
being inherited by the lower level class.

7. (Original) The hierarchical class architecture of claim 6, further comprising:

an additional attribute, wherein
the additional attribute is assigned to the lower level class, and

the attribute describes an object in the lower level class.

8. (Original) The hierarchical class architecture of claim 6, wherein the attribute comprises a distinctive domain value set.

9. (Original) The hierarchical class architecture of claim 6, further comprising:

a class, wherein

the class is below the lower level class in the hierarchical class architecture, and
the class is configured to inherit the attribute.

10. (Original) The hierarchical class architecture of claim 6, wherein the lower level class is configured to be expanded horizontally by virtue of being configured to provide for addition of a class, and the class is configured to inherit the attribute.

11. (Currently Amended) A computer system comprising:
a processor;
a computer readable medium coupled to the processor; and
computer code, encoded in the computer readable medium, configured to cause the processor to:

set a class hierarchy, wherein

the class hierarchy comprises an upper level class and a lower level class,
and
the objects are members of at least one of the upper level class and the lower level class;

assign an attribute to the **[[top]]** upper level class, wherein the attribute describes the objects; and

provide inheritance of the attribute by the lower level class, wherein

the attribute is within a first domain with regard to the upper level class,
the attribute is within a second domain with regard to the lower level class,

a second domain value set of the second domain is smaller than a first domain value set of the first domain, and
the attribute is restricted to the second domain value set upon the inheritance of the attribute by the lower level class.

12. (Original) The computer system of claim 11, wherein the computer code is further configured to cause the processor to:

assign an attribute to the lower level class, the attribute describing an object that is a member of the lower level class.

13. (Original) The computer system of claim 11, wherein the attribute comprises a distinctive domain value set.

14. (Original) The computer system of claim 11, wherein the class hierarchy further comprises a class below the lower level class in the class hierarchy, and the computer code is further configured to cause the processor to:

provide inheritance of the attribute by the class.

15. (Original) The computer system of claim 11, wherein the computer code is further configured to cause the processor to:

expand the class hierarchy horizontally by adding a class to the lower level class;

and

provide inheritance of the attribute by the class.

16. (Currently Amended) An apparatus for arranging objects comprising:
means for setting a class hierarchy, wherein

the class hierarchy comprises an upper level class and a lower level class,

and

the objects are members of at least one of the upper level class and the

lower level class;

means for assigning an attribute to the **[[top]]** upper level class, wherein the attribute describes the objects; and

means for inheriting of the attribute by the lower level class, wherein
the attribute is within a first domain with regard to the upper level class,
the attribute is within a second domain with regard to the lower level class,
a second domain value set of the second domain is smaller than a first
domain value set of the first domain, and
the attribute is restricted to the second domain value set by the means for
inheriting.

17. (Currently Amended) The apparatus of claim 16, further comprising:
means for superceding said attribute of said upper level class comprising means
for assigning an attribute to the lower level class, the attribute describing
an object that is a member of the lower level class.

18. (Original) The apparatus of claim 16, wherein the attribute comprises a
distinctive domain value set.

19. (Original) The apparatus of claim 16, wherein the class hierarchy further
comprises a class below the lower level class in the class hierarchy, and further
comprising:

means for inheriting of the attribute by the class.

20. (Original) The apparatus of claim 16, further comprising:
means for expanding the class hierarchy horizontally by adding a class to the
lower level class; and
means for inheriting of the attribute by the class.

21. (Currently Amended) A computer program product, encoded in computer
readable media, comprising:

a first set of instructions, executable on a computer system, configured to set a
class hierarchy, wherein
the class hierarchy comprises an upper level class and a lower level class,
and

the objects are members of at least one of the upper level class and the lower level class;

a second set of instructions, executable on the computer system, configured to assign an attribute to the **[[top]]** upper level class, wherein the attribute describes the objects; and

a third set of instructions, executable on the computer system, configured to provide inheritance of the attribute by the lower level class, wherein the attribute is within a first domain with regard to the upper level class, the attribute is within a second domain with regard to the lower level class, a second domain value set of the second domain is smaller than a first domain value set of the first domain, and the attribute is restricted to the second domain value set by the third set of instructions.

22. (Currently Amended) The computer program product of claim 21, further comprising:

a fourth set of instructions, executable on the computer system, configured to supercede said attribute of said upper level class by virtue of being configured to assign an attribute to the lower level class, the attribute describing an object that is a member of the lower level class.

23. (Original) The computer program product of claim 21, wherein the attribute comprises a distinctive domain value set.

24. (Original) The computer program product of claim 21, wherein the class hierarchy further comprises a class below the lower level class in the class hierarchy, and further comprising:

a fourth set of instructions, executable on the computer system, configured to provide inheritance of the attribute by the class.

25. (Original) The computer program product of claim 21, further comprising:

a fourth set of instructions, executable on the computer system, configured to expand the class hierarchy horizontally by adding a class to the lower level class; and

a fifth set of instructions, executable on the computer system, configured to provide inheritance of the at of the attribute by the class.

26. (Currently Amended) The method of arranging objects of claim 1, further comprising:

associating the upper level class with [[a]] the first domain value set, and associating the lower level class with [[a]] the second domain value set.

27. (Currently Amended) The method of arranging objects of claim 26, wherein

another attribute is within [[a]] the second domain ~~with regard to the lower level~~ class.

28. (Previously Presented) The method of arranging objects of claim 27, wherein

the another attribute is an overriding attribute.

29. (Previously Presented) The method of arranging objects of claim 27, further comprising:

superceding the attribute with the another attribute, wherein the superceding is performed if the second domain is different from the first domain.

30. Cancelled.

31. (Previously Presented) The method of arranging objects of claim 1, further comprising:

associating an item to a class within the class hierarchy such that all entries of a record of the item are non-empty.

32. Cancelled.

33. (Currently Amended) The hierarchical class architecture of objects of claim ~~[[32]]~~ 6, further comprising:
another attribute, wherein the another attribute is another attribute within ~~[[a]]~~ the
second domain ~~with regard to the lower level class.~~

34. (Previously Presented) The hierarchical class architecture of objects of claim 33, wherein
the another attribute is an overriding attribute.

35. (Previously Presented) The hierarchical class architecture of objects of claim 33, wherein
the another attribute is configured to supercede the attribute, and
the another attribute supercedes the attribute if the second domain is different
from the first domain.

36. Cancelled.

37. (Previously Presented) The hierarchical class architecture of objects of claim 6, further comprising:
an item having a record with a plurality of entries, wherein
the item is associated to a class within the class hierarchy such that the
entries are non-empty.

38. (Currently Amended) The computer system of ~~arranging objects of~~ claim 11, wherein the computer code is further configured to cause the processor to:
associate the upper level class with ~~[[a]]~~ the first domain value set, and
associate the lower level class with ~~[[a]]~~ the second domain value set.

39. (Currently Amended) The computer system of ~~arranging objects of~~ claim 38, wherein

another attribute is within [[a]] the second domain ~~with regard to the lower level~~
class.

40. (Currently Amended) The computer system of ~~arranging objects of~~ claim 39, wherein
the another attribute is an overriding attribute.

41. (Currently Amended) The computer system of ~~arranging objects of~~ claim 39, wherein the computer code is further configured to cause the processor to:
supercede the attribute with the another attribute, if the second domain is different
from the first domain.

42. Canceled

43. (Currently Amended) The computer system of ~~arranging objects of~~ claim 11, wherein the computer code is further configured to cause the processor to:
associate an item to a class within the class hierarchy such that all entries of a
record of the item are non-empty.

44. (Currently Amended) The ~~computer system of arranging objects~~
apparatus of claim 16, wherein the computer code is further configured to cause the
processor to:

associate the upper level class with [[a]] the first domain value set, and
associate the lower level class with [[a]] the second domain value set.

45. (Currently Amended) The apparatus of ~~arranging objects of~~ claim 44,
wherein
another attribute is within [[a]] the second domain ~~with regard to the lower level~~
class.

46. (Currently Amended) The apparatus of ~~arranging objects of~~ claim 45,
wherein
the another attribute is an overriding attribute.

47. (Currently Amended) The apparatus of ~~arranging objects of~~ claim 45, further comprising:

means for superceding the attribute with the another attribute, wherein
the superceding is performed if the second domain is different from the
first domain.

48. Cancelled.

49. (Currently Amended) The apparatus of ~~arranging objects of~~ claim 16, further comprising:

means for associating an item to a class within the class hierarchy such that all
entries of a record of the item are non-empty.

50. (Currently Amended) The computer program product of claim 21, further comprising:

~~means for associating~~ a fourth set of instructions, executable on the computer
system, configured to associate the upper level class with [[a]] the first
domain value set[[,]] ; and

~~means for associating~~ a fifth set of instructions, executable on the computer
system, configured to associate the lower level class with [[a]] the second
domain value set.

51. (Currently Amended) The computer program product of claim 50,
wherein

another attribute is within [[a]] the second domain ~~with regard to the lower level
class.~~

52. (Previously Presented) The computer program product of claim 51,
wherein

the another attribute is an overriding attribute.

53. (Currently Amended) The computer program product of claim 51, further

comprising:

~~means for superceding~~ a sixth set of instructions, executable on the computer system, configured to supercede the attribute with the another attribute, if the second domain is different from the first domain.

54. Cancelled.

55. (Currently Amended) The computer program product of claim 21, further comprising:

~~means for associating~~ a fourth set of instructions, executable on the computer system, configured to associate an item to a class within the class hierarchy such that all entries of a record of the item are non-empty.